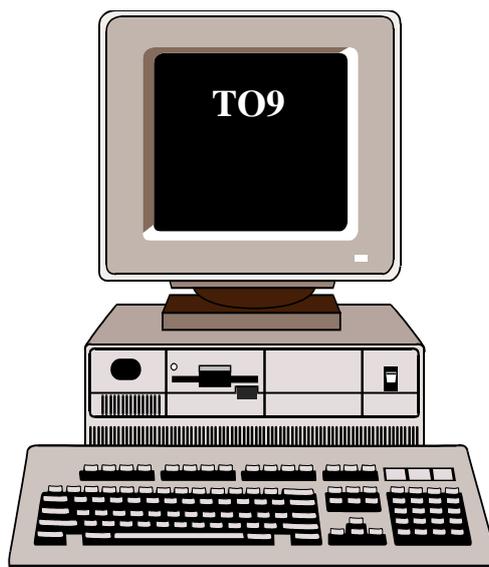


LES MODES D'ADRESSAGE (6809)

TRAVAUX PRATIQUES (TP₁+ TP₂)

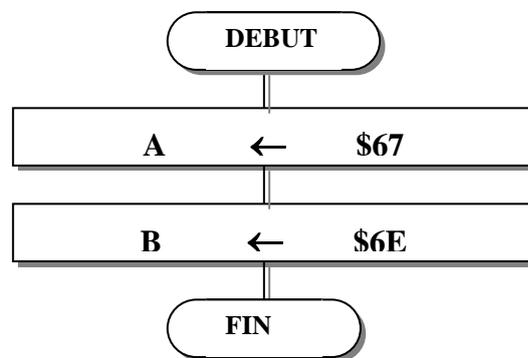


TRAVAIL A FAIRE :

1. Complétez la feuille de programmation ainsi que l'ordinogramme correspondant.
2. Calculer le temps d'exécution du programme.
3. Saisir le programme.
4. Justifier tous les calculs.

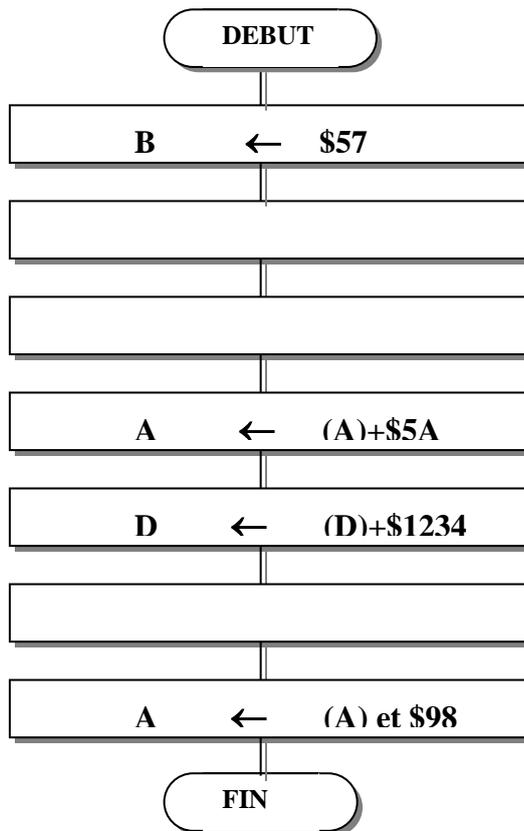
Conventions :

A	Registre A
(A)	Contenu du registre A
\$8000	Adresse \$8000
(\$8000)	Contenu de l'adresse \$8000

A. L'ADRESSAGE IMMEDIAT :Exercice 1 :

PROGRAMME SOURCE (Langage assembleur)				Adresse	Nbre de cycle	Registre		PROGRAMME OBJET (langage machine)			
Etiquette	Mnémonique	Opérande	Commentaires			A	B	1 ^{er} octet	2 ^{ème} octet	3 ^{ème} octet	4 ^{ème} octet
	ORG	\$B000									
DEBUT				\$B000							

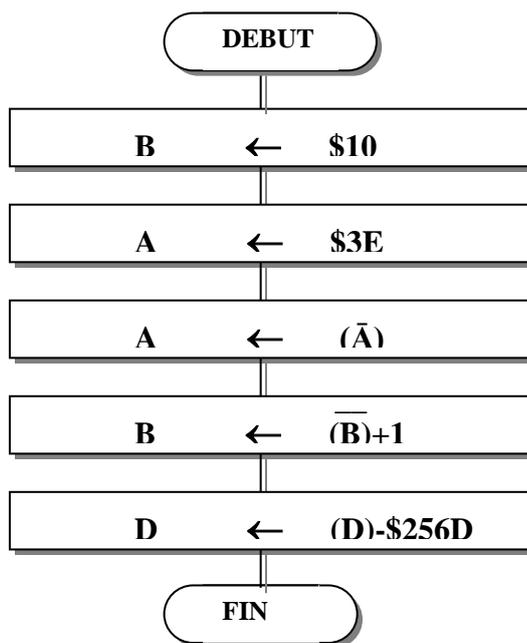
Exercice 2 :



PROGRAMME SOURCE (Langage assembleur)				Adresse	Nbre de cycle	Registre		PROGRAMME OBJET (langage machine)			
Etiquette	Mnémonique	Opérande	Commentaires			A	B	1 ^{er} octet	2 ^{ème} octet	3 ^{ème} octet	4 ^{ème} octet
	ORG	\$B000									
DEBUT				\$B000							
	LDA	#\$87									
	SUBB	#\$28									
							C4	98			

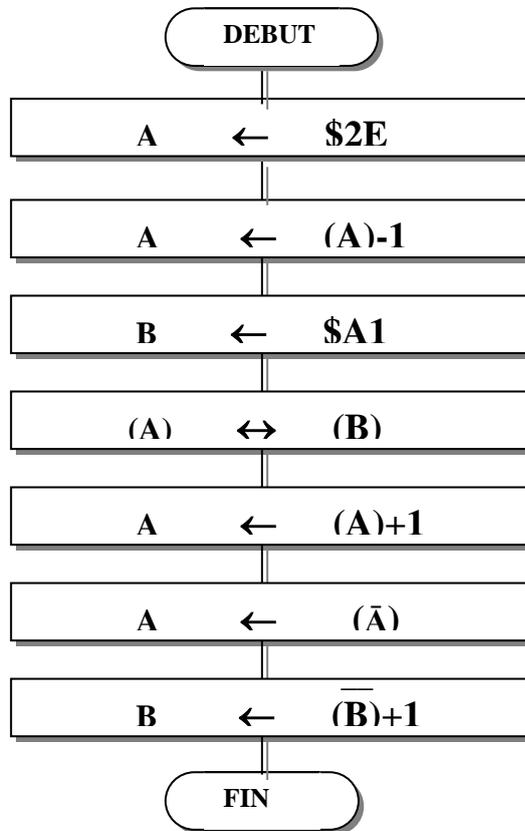
B. L'ADRESSAGE INHERENT ET IMMEDIAT :

Exercice 1 :



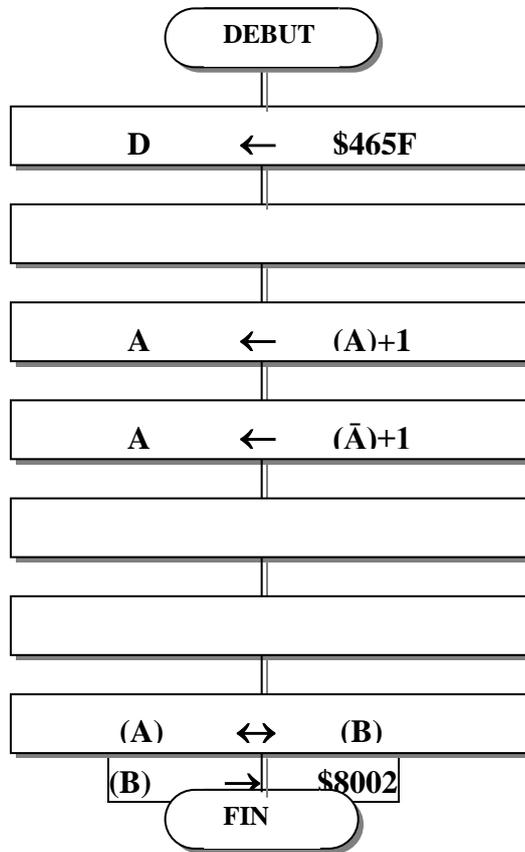
PROGRAMME SOURCE (Langage assembleur)				Adresse	Nbre de cycle	Registre		PROGRAMME OBJET (langage machine)			
Etiquette	Mnémonique	Opérande	Commentaires			A	B	1 ^{er} octet	2 ^{ème} octet	3 ^{ème} octet	4 ^{ème} octet
				\$B000							

Exercice 2 :



PROGRAMME SOURCE (Langage assembleur)				Adresse	Nbre de cycle	Registre		PROGRAMME OBJET (langage machine)			
Etiquette	Mnémonique	Opérande	Commentaires			A	B	1 ^{er} octet	2 ^{ème} octet	3 ^{ème} octet	4 ^{ème} octet
				\$B000							

Exercice 4 :



PROGRAMME SOURCE (Langage assembleur)				Adresse	Nbre de cycle	Registre		PROGRAMME OBJET (langage machine)			
Etiquette	Mnémonique	Opérande	Commentaires			A	B	1 ^{er} octet	2 ^{ème} octet	3 ^{ème} octet	4 ^{ème} octet
DEBUT				\$B000							
	TFR	B, A									
	CLRB						C6	47			

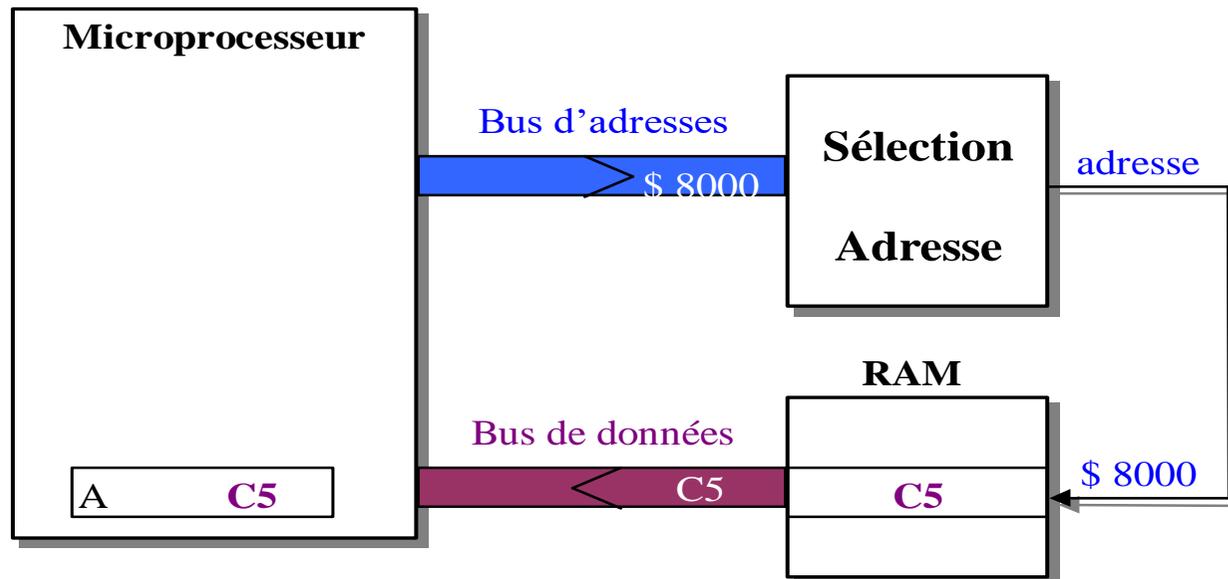
C. L'ADRESSAGE ETENDU :

Exemple :

LDA >\$8000

Chargement de l'accumulateur A avec le contenu de l'adresse \$8000

Mémoire	
FFFF	
B002	00
B001	80
B000	B6
8000	C5
0000	



Après exécution de l'instruction, l'accumulateur A contient la valeur \$ C5

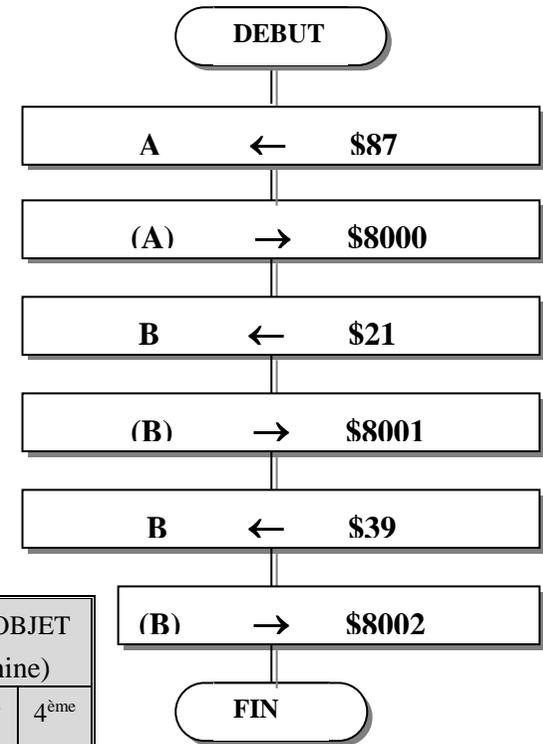
Exercice 1 :

Relever le contenu des adresses précisées :

Pour cela, après avoir avancé d'un pas (# T) dans le programme.

Taper # D\$8000/ noter alors le contenu des adresses.

Puis continuer l'exécution du mode pas à pas (# T).



PROGRAMME SOURCE (Langage assembleur)				Adresse	Nbre de cycle	Registre ou Adresse					PROGRAMME OBJET (langage machine)			
Etiquette	Mnémonique	Opérande	Commentaires			A	B	8000	8001	8002	1 ^{er} octet	2 ^{ème} octet	3 ^{ème} octet	4 ^{ème} octet
				\$B000										

Exercice 4 :

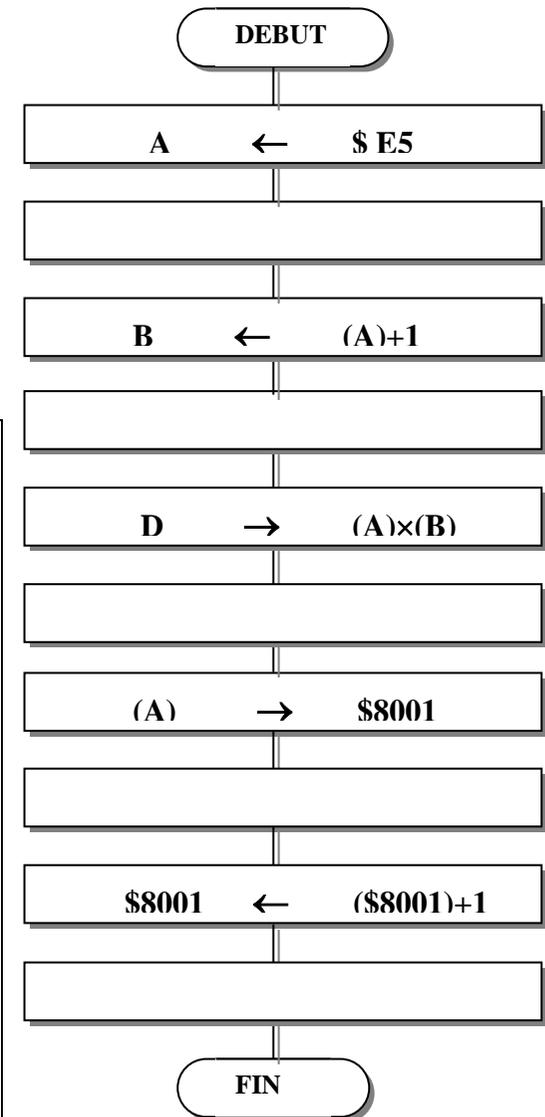
Relever le contenu des adresses précisées :

Pour cela, après avoir avancé d'un pas (# T) dans le programme.

Taper # D\$8000/ noter alors le contenu des adresses.

Puis continuer l'exécution du mode pas à pas (# T).

PROGRAMME SOURCE (Langage assembleur)				Adresse	Nbre de cycle	Registre ou Adresse				PROGRAMME OBJET (langage machine)			
Etiquette	Mnémonique	Opérande	Commentaires			A	B	8000	8001	1 ^{er} octet	2 ^{ème} octet	3 ^{ème} octet	4 ^{ème} octet
				\$B000									
	TFR	A,B											
	COMB												
	STA	>\$8000											
	DEC	>\$8000											
	LDD	>\$8000											



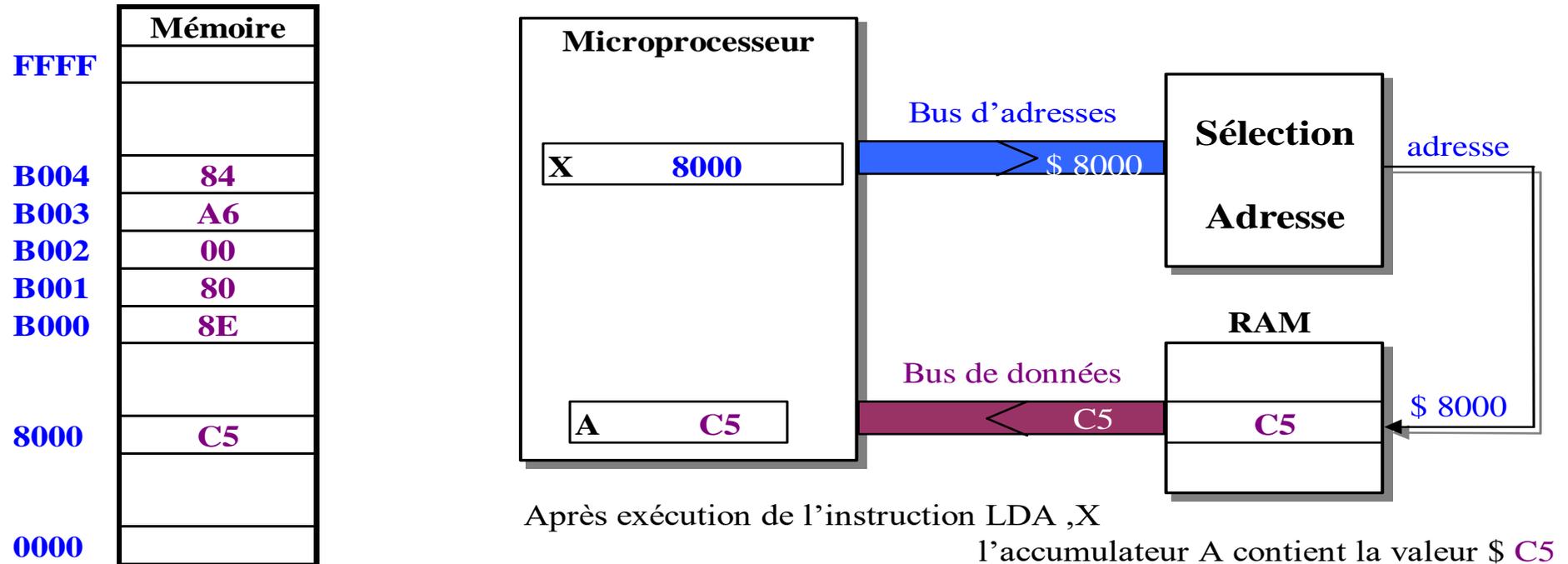
D. L'ADRESSAGE INDEXE :

Exemple :

LDX #8000

LDA ,X

Chargement de l'accumulateur A avec la valeur située à l'adresse pointée par le registre X



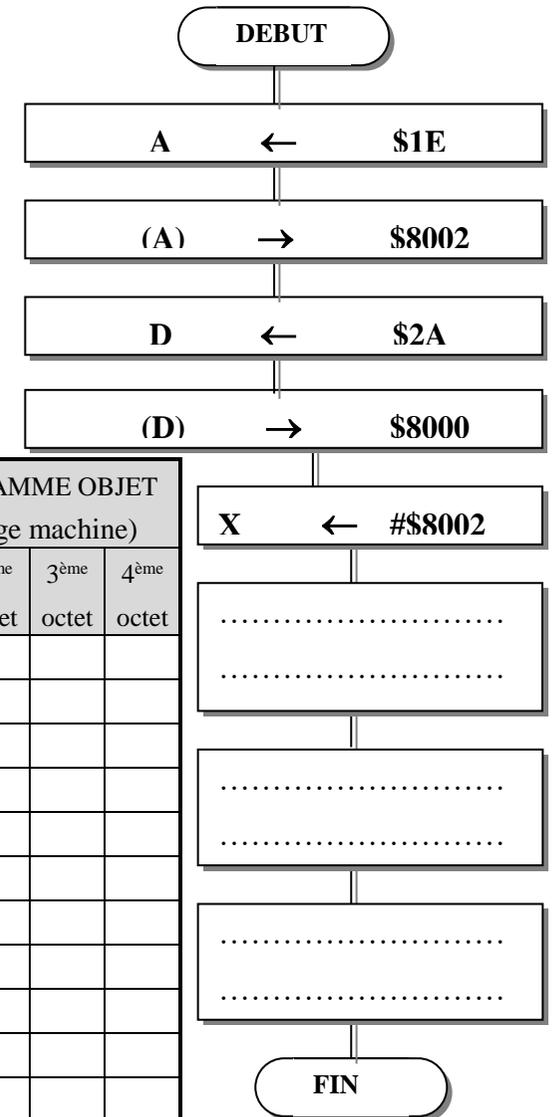
Exercice :

Relever le contenu des adresses précisées :

Pour cela, après avoir avancé d'un pas (# T) dans le programme.

Taper # D\$8000/ noter alors le contenu des adresses.

PUIS CONTINUER L'EXECUTION DU MODE PAS A PAS (# T).



PROGRAMME SOURCE (Langage assembleur)				Adresse	Nbre de cycle	Registre ou Adresse					PROGRAMME OBJET (langage machine)			
Etiquette	Mnémonique	Opérande	Commentaires			A	X	8002	8001	8000	1 ^{er} octet	2 ^{ème} octet	3 ^{ème} octet	4 ^{ème} octet
				\$B000										
	LDB	,-X												
	LDB	,-X												
	LDB	,X++												